Docket JP920010326US1

Appl. No.: 10/736,343
Filed: December 15, 2003

IN THE CLAIMS

Please cancel claims 1-25 and enter new claims as follows.

Claims 1-25 (canceled)

26. (new) A method for executing, by a processor of a computer system, a set of program instructions for a loop, wherein the method comprises the steps of:

associating a unique proxy value with each indirect loop index variable of the loop, wherein each unique proxy value is a different prime number;

calculating, for each iteration of the loop, an indirectly indexed access pattern based upon the unique values;

determining whether cross-iteration dependencies exist between any two iterations of the loop based upon the indirectly indexed access patterns of the two iterations;

scheduling the program instructions of the loop across iterations into waves based on the cross-iteration dependencies found;

and executing the waves.

27. (new) The method of claim 26, wherein the indirectly indexed access pattern for each iteration is calculated by forming the product of the unique proxy values associated with each of the indirect loop index variables on a decision path of the loop for that iteration.

28. (new) The method of claim 27, wherein the determining of cross-iteration dependencies is accomplished by finding the greatest common divisor between two indirectly indexed access patterns, wherein a greatest common divisor of 1 indicates that no dependencies exist between the two respective iterations.

29. (new) The method of claim 27, wherein an indirect indexed access pattern is calculated for each possible decision path of the loop.

30. (new) The method of claim 26, the indirectly indexed access patterns for the respective iterations of the loop have pattern values, and wherein for each iteration the pattern values of the indirectly indexed access pattern do not exceed three in number regardless of how many statements are in the loop.

31. (new) A computer program product for executing, by a processor of a computer system, a set of program instructions for a loop, the computer program product comprising computer software stored on a tangible, computer-readable storage medium for performing the steps of:

associating a unique proxy value with each indirect loop index variable of the loop, wherein each unique proxy value is a different prime number;

calculating, for each iteration of the loop, an a plurality of indirectly indexed access patterns based upon the unique values;

determining whether cross-iteration dependencies exist between any two iterations of the loop based upon the indirectly indexed access patterns of the two iterations; and

scheduling the program instructions of the loop across iterations into waves based on the cross-iteration dependencies found, wherein the waves are executed responsive to the scheduling.

32. (new) The computer program product of claim 31, wherein the indirectly indexed access patterns for each iteration is are calculated by forming the product of the unique proxy values associated with each of the indirect loop index variables on a decision path of the loop for that iteration.

33. (new) The computer program product of claim 32, wherein the determining of cross-iteration dependencies is accomplished by finding the greatest common divisor between two indirectly indexed access patterns, wherein a greatest common divisor of 1 indicates that no dependencies exist between the two respective iterations.

34. (new) The computer program product of claim 32, wherein an indirect indexed access pattern is calculated for each possible decision path of the loop.

35. (new) The computer program product of claim 31, the indirectly indexed access patterns for the respective iterations of the loop have pattern values, and wherein for each iteration the pattern values of the indirectly indexed access pattern do not exceed three in number regardless of how many statements are in the loop.

36. (new) A computer system having program instructions stored on a computer-readable medium for executing, by a processor of the computer system, a set of the program instructions for a loop, wherein the executing comprises performing the steps of:

associating a unique proxy value with each indirect loop index variable of the loop, wherein each unique proxy value is a different prime number;

calculating, for each iteration of the loop, an a plurality of indirectly indexed access patterns based upon the unique values;

determining whether cross-iteration dependencies exist between any two iterations of the loop based upon the indirectly indexed access patterns of the two iterations; and

scheduling the program instructions of the loop across iterations into waves based on the cross-iteration dependencies found, wherein the waves are executed responsive to the scheduling.

37. (new) The computer program product of claim 36, wherein the indirectly indexed access patterns for each iteration is are calculated by forming the product of the unique proxy values associated with each of the indirect loop index variables on a decision path of the loop for that iteration.

38. (new) The computer program product of claim 37, wherein the determining of cross-iteration dependencies is accomplished by finding the greatest common divisor between two indirectly indexed access patterns, wherein a greatest common divisor of 1 indicates that no dependencies exist between the two respective iterations.

39. (new) The computer program product of claim 37, wherein an indirect indexed access pattern is calculated for each possible decision path of the loop.

40. (new) The computer program product of claim 36, the indirectly indexed access patterns for the respective iterations of the loop have pattern values, and wherein for each iteration the pattern values of the indirectly indexed access pattern do not exceed three in number regardless of how many statements are in the loop.